Title: NNSA/CEA Workflow Workshop Report

Author(s): Randles, Timothy C.
Friedman-Hill, Ernest
Laney, Dan
Capul, Julien

Intended for: Report

Issued: 2021-05-27 (rev.1)

NNSA/CEA Workflow Workshop Report 2021

Organizers: Julien Capul (CEA), Ernest Friedman-Hill (SNL), Dan Laney (LLNL), Tim Randles (LANL)

Participants:

- CEA: Francis Belot, Julien Capul, Olivier Delhomme, Marie-Pierre Oudot, Bruno Roche, Aline Roy
- LANL: Tim Germann, Patricia Grubel, Scot Halverson, Christoph Junghans, Robert Pavel, Tim Randles, Brad Settlemyer
- LLNL: Dong Ahn, James Corbett, Frank Di Natale, David Dawson, David Domyancic, Charles Doutriaux, Andrew Fillmore, Becky Haluska, Stephen Herbein, Dan Laney, Katie Lewis, Alex Murray, Esteban Pauli, Luc Peterson
- SNL: Brett Clark, Ernest Friedman-Hill, George Orient

Executive Summary

Researchers from three NNSA labs plus CEA recently met for a half-day workshop on scientific and engineering workflows in March of 2021. Tools, projects and use cases from each institution were described. A wide range of unique capabilities and requirements were represented. The workshop highlights the fact that the CEA/NNSA workflows space mirrors the broader open science workflows community, with multiple technologies under development in a number of science domains and under a number of funding streams, with differing capabilities and focuses. Despite the number of tools, presentations and discussions have shown that these tools cover specific mission spaces at the four labs, each with distinctive capabilities that do not completely overlap with each other. We believe there is a strong interest in the short term for sharing lessons learned and collaborating on benchmarks and site evaluation of projects.

This report, prepared by the NNSA/CEA Workflows Working Group, briefly summarizes the presentations in the areas of domain specific workflows, end user environments, data management, job and resource management, and infrastructure, and then identifies six broad areas for potential collaboration. A key finding is that users could benefit from greater interoperability, compatibility, and composability of the workflow technologies under development and that point-to-point collaboration opportunities should be identified to explore these aspects.

Introduction

High performance scientific and engineering workflows are a very active area of work at the NNSA and CEA laboratories. It is recognized that opportunities may exist for greater sharing and collaboration between the many projects underway. A workshop for the four labs was

proposed as a way to bring together the workflow communities to share details about their projects.

The goal of this workshop is to foster discussion of the various workflow projects underway at the labs and find places where we can collaborate and share, producing a NNSA/CEA workflow strategy document we can socialize with our program management.

Process

The NNSA/CEA Workflows working group (POCs: Julien Capul, Ernest 'Foss' Friedman-Hill, Dan Laney, Tim Randles) held a half day workshop on February 2, 2021, inviting members of the workflow community at each lab. Each of the four laboratories had one hour to present their work on workflows. The scope of these presentations was to discuss their workflow projects, tools, use cases, and strengths and weaknesses of these activities. These presentations described a wide variety of projects, from general workflow managers to simulation data management tools to tools tailored for specific styles of workflows such as large ensembles. The number and diversity of the presentations is evidence of the importance of workflows to current and future high performance scientific computing.

Each lab's slides were distributed after the February 2 meeting. The labs then held discussions about the work of their peer institutions and identified areas for future collaboration. On February 9 a second meeting was held to discuss the proposed collaborations.

Overview of Workflow Activities

The four laboratories created a summary spreadsheet containing sixteen projects. Most of these were represented in the meetings, although a few (e.g., Dakota) are only represented in the summary spreadsheet with documentation available in other locations. In this section we provide a high level summary of the various projects and efforts:

Workflow Management Systems/Libraries (WMSs): we discussed five internally developed workflow management systems, and include information on two additional systems developed in the Office of Science that are part of the ECP ExaWorks software development kit. The NNSA/CEA workflow tools landscape mirrors the broader open science workflows community, with multiple technologies under development in a number of science domains and under a number of funding streams, with differing capabilities and focuses. A brief summary is provided here:

- **Themis:** highly focused ensemble manager for large scale simulation workloads that leverages existing user batch-scripts with minimal changes
- **Maestro:** flexible, serverless orchestration of workflows represented as directed acyclic graphs described in a structured and readable YAML specification
- **Merlin:** highly scalable distributed execution of task graphs coordinated by a persistent server, consumes Maestro YAML specs

- **BEEFlow:** container-oriented distributed workflow system that can run workflows on HPC and clouds described by the Common Workflow Language
- **NextGen Workflow:** flexible workflow management with a graphical user interface for building and running workflows that span desktop applications and HPC jobs
- **Parsl (ExaWorks SDK member):** a Python parallel programming library that schedules and executes dynamic task graphs within batch allocations
- **RADICAL Cybertools (ExaWorks SDK member):** a set of middle-ware components for workflows, encompassing an abstraction layer on top of grid and batch schedulers, and tools for managing ensembles of simulations

Each tool provides a set of trade-offs and functionality that enables a wide-variety of use-cases, from 'legacy' simulation studies, through UQ analysis, to integrated ML + HPC applications to be performed. For example, multiple workflow descriptions are supported: existing batch scripts which have low barrier to entry but limited flexibility, Maestro YAML which is designed to be succinct and easily support parameter studies, Common Workflow Language and NGW's IWF format, which are very general but can be more verbose and complex, and Python APIs which provide flexibility but with the complexity of being dynamic programs. Similarly, several tools run completely in user-space and do not require persistent services which makes them more portable but also less flexible. In general, the developers saw potential in discussing collaborations centered on compatibility/conversion of workflow specifications as well as exploring ways in which the tools could be composed or interoperate.

Domain Specific Workflows and End User Environments:

- **Dakota:** a front-end for design optimization and parametric analysis, often leveraged in combination with back-end workflow engines or directly layered on existing batch systems
- **PLATO:** topology-optimization based design environment, which provides problem setup and job management tools, as well as an optimization engine
- **PVCS:** a test engine for running large test suites scalably on HPC systems
- **ELODI:** combines workflow management and data management in a user-focused problem solving environment
- **Sandia Analysis Workbench:** graphical interface combining problem setup and meshing, simulation management and analysis, with integrated Simulation Data Management (SDM)

In general, these tools are driven directly by domain-specific requirements, and often embed general workflow managers (e.g., NGW in SAW and ELODI). The community agreed that these teams can benefit from greater flexibility in underlying tools, such as the ability to compose tools to achieve the best possible performance and reliability for a given workflow use-case.

Data Management:

The projects presented address the need for standardized data ingestion and query capabilities while ensuring backend flexibility and scalability, for tools to manipulate workflow digital artifacts and increase end-user productivity and for multi-user data persistent services to foster collaboration:

- **Sina:** provides a hierarchical JSON (or HDF5) schema for simulation meta-data and non-bulk data (scalars, curves, …), a C++ library for writing the data, and a Python package for ingesting and querying the data in databases.
- **Kosh:** leverages Sina as a data catalogue, and adds file movement/management, transparent file readers (into standard Python data structures like Numpy and Pandas), and utilities for machine learning workflows (e.g., sub-sampling, data subsetting for cross-validation).
- **DMTCP (distributed multithreaded checkpointing):** arbitrarily-timed checkpointing of an application's memory footprint, in user-space. Enables checkpoint-restart of applications that do not provide native checkpoint support.

Job and resource management:
These projects addresses the need for efficient multi-cluster orchestration of workflow tasks, for high-throughput job scheduling to enable ensemble studies, and for tasks co-scheduling adapted to in situ workflows:

- **Flux:** a next-generation resource and job management framework that expands the scheduler's view beyond the single dimension of "nodes." Instead of simply developing a replacement for SLURM and Moab, Flux offers a framework that enables new resource types, schedulers, and framework services to be deployed as data centers continue to evolve.
- **BEEFlow:** provides the ability to execute workflow tasks across multiple HPC and cloud resources. This includes being able to start and stop HPC jobs, as well as the ability to create and delete virtual machines (VMs) running on multiple public and private cloud platforms.
- **KMS:** provides a unified interface for end users to launch and manage computing jobs across multiple clusters and provides extended scheduling capabilities to system administration to optimize production loads on machines.

Infrastructure: while not a project per-se, the community spent a significant amount of time discussing the ways in which the infrastructure supported by our data centers can enhance user workflow capabilities, particularly in the areas of scalable persistent services and increasing the portability of workflows by supporting a standard set of services that workflow teams can leverage. Many workflow management systems and workflows use services that do not fit well on traditional HPC systems. Examples of these services include key-value stores (etcd), data-structure stores (REDIS), message queueing services (RabbitMQ, Kafka), and NoSQL databases (Cassandra, MongoDB). The ability to deploy and manage these services on appropriate resources, such as an on-premise cloud or container orchestration service, is a key capability.

Possible Areas for Future Collaboration

The discussions at each site, and the combined discussion on day two highlighted several areas where the labs can collaborate. The plan for facilitating collaboration, information sharing, and enhanced capability is three fold:

1. Identify point-to-point collaborations between teams
2. Identify topics that could be discussed in-depth at future meetings
3. Provide clear recommendations as a community to NNSA/CEA facilities on capabilities required for future workflow environments

Composition: the ability to compose multiple workflow systems is a useful way to share technologies. Particularly for the domain-specific or user-centric systems that encompass both workflow management, product lifecycle management, or higher level abstractions (e.g., optimization). In these cases, implementing lower level functionality on existing tools that have been proven to scale can be an asset to teams that would otherwise implement bespoke solutions. Specific examples include:

1. Leveraging Themis as a low-level ensemble execution node in BEEflow or NGW
2. Leveraging nested Maestro DAG's inside other tools

Compatibility: the ability to convert a workflow implemented in one system to another was called out as a potential collaboration point. Themis leverages an implied workflow, Maestro/Merlin leverages YAML specs, BEEFlow and NextGen Workflow leverage the Common Workflow Language or XML respectively. In all cases, it is theoretically possible to convert one workflow representation to another, assuming the execution models can be converted (e.g., shell script vs. container invocation). The community thought a topic for future discussions would be the feasibility of increasing portability through conversion.

Interoperability: nearly all workflow systems separate the specification of the workflow and its representation as a task-graph or DAG, analogous to the way in which a compiler converts a human readable input into an abstract syntax tree. All HPC workflow managers support some level of abstraction over the lower level system schedulers to aid portability. A research topic is to investigate how various levels of abstraction in workflow systems could be leveraged to produce interoperability, such as the ability to leverage multiple executor levels from a common workflow description. In this area, community engagement is key, as evidenced by the ExaWorks (ECP) / WorkflowsRI (NSF) summit in which 27 workflow management systems were represented in discussions on various topics. For this area, concrete next steps include discussions on the ExaWorks J/PSI portable job scheduling interface, as well as community discussions on functional decomposition of workflow systems.

Ex-situ and In-Situ Workflows: this topic is focused on ways in which a single workflow graph can span activities within a single code and connect them to other tools using a common graph representation. For example, specifying an in-situ data analysis pipeline to be executed at each time step, or uniformly representing model assembly, a multistep analysis, and postprocessing phases, in which the number of graph nodes does not correspond one-to-one with executable processes but rather with conceptual processing steps.

Infrastructure: workflow systems always operate in some larger software and hardware environment, and the ability to depend on existing infrastructure elements can make a workflow system simpler and more robust. Developing abstractions or standards for infrastructure

elements would strengthen our systems and simplify sharing workflows and standing up systems at new sites. Defining a set of “always-on” services and a standard way of accessing them will enhance workflow and workflow management system portability across sites. Developing a common method for using tools like GitLab to implement continuous integration (CI) and testing on HPC platforms will also improve the portability of applications across sites.

Data Management: A range of data management needs are evident, from data transfer to short-term storage to archiving and retrieval. Performance, portability, and security are all areas of common interest.

- LANL will evaluate Sina and Kosh for data management that will enhance the ability to capture workflow meta-data in a portable format
- LLNL will explore the inclusion BEEflow in the ExaWorks SDK, particularly BEE’s ability to scale out tests to do performance testing beyond what is typically achievable from Travis and related tools

Conclusion

The goal of the workshop was to foster discussion and awareness on the various software development projects and use cases related to simulation workflow and data management at the four labs. The workshop brought together a community of 35 developers and practitioners from the four labs, with 15 different speakers presenting their projects and use cases. The report provides an overview of these technical activities underway. Despite the number of tools, presentations and discussions have shown that these tools cover specific mission spaces at the four labs, each with distinctive capabilities that do not completely overlap with each other. We believe there is a strong interest in the short term for sharing lessons learned and collaborating on benchmarks and site evaluation of projects. But discussions have also highlighted the need for further research and collaboration to drive this set of projects towards more interoperability with the long term goal to provide an ecosystem of tools and services that workflow systems developers and practitioners can use and compose to expand simulation capabilities.

Our next steps as a community will be to raise awareness of these workflow tools at our labs and organize workflow training and breakout sessions at appropriate meetings.

Acknowledgements

This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001).
LA-UR-21-23758

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Appendix: Table Summarizing Projects and Tools

Project Name	Lab	Contact email	Brief Summary	Workflow, Data, or Job Management
BEE	LANL	Tim Randles trandles@lanl.gov	Distributed workflow management system capable of running tasks on HPC clusters and clouds. Uses CommonWorkflowLanguage for workflow definition. Supports Charliecloud and Singularity container runtimes, Slurm and LSF resource managers. Support for AWS, OpenStack, and Google Cloud Platform under development.	Workflow and job manager
Dakota	Sandia	https://dakota.sandia.gov/resources.html	The Dakota software's advanced parametric analyses enable design exploration, model calibration, risk analysis, and quantification of margins and uncertainty with computational models. The Dakota toolkit provides a flexible, extensible interface between such simulation codes and its iterative systems analysis methods, which include: optimization with gradient and nongradient-based methods; uncertainty quantification with sampling, reliability, stochastic expansion, and epistemic methods; parameter estimation using nonlinear least squares (deterministic) or Bayesian inference (stochastic); and sensitivity/variance analysis with design of experiments and parameter study methods.	Workflow

			These capabilities may be used on their own or as components within advanced strategies such as hybrid optimization, surrogate-based optimization, mixed integer nonlinear programming, or optimization under uncertainty.	
Distributed Multi-threaded Checkpointing (DMTCP)	LANL	Scot Halverson sah@lanl.gov	DMTCP transparently checkpoints a single-host or distributed computation in user-space -- with no modifications to user code or to the O/S. It works on most Linux applications, including Python, Matlab, R, GUI desktops, MPI, etc. It is robust and widely used. Among the applications supported by DMTCP are MPI (various implementations), OpenMP, MATLAB, Python, Perl, R, and many programming languages and shell scripting languages. With the use of TightVNC, it can also checkpoint and restart X-Window applications. The OpenGL library for 3D graphics is supported through a special plugin. It also has strong support for HPC (High Performance Computing) environments, including MPI, SLURM, InfiniBand, and other components. DMTCP supports the commonly used OFED API for InfiniBand, as well as its integration with various implementations of MPI, and resource managers (e.g., SLURM).	Job/data management
Elodi	CEA	Julien Capul julien.capul@cea.fr	Elodi Project aims at providing a simulation workflow and data management system to CEA/DAM physicists. The workflow management system is based on an adapted version of Sandia Next Generation Workflow software (NGW). The data management system will be a custom solution built on top of Git and Gitlab for non-bulk data.	Workflow and data management
Flux	LLNL	Dong Ahn ahn1@llnl.gov	Flux is a next-gen scheduler framework. Flux is hierarchical and provides API's for job coordination and communication in addition to a sophisticated scheduler API. Flux is highly scalable and can run in user-space, providing a common API	Job management, Key value store, pub/sub communicati

			substrate on which to build portable workflows. https://computing.llnl.gov/projects/flux-building-framework-resource-management	on
KMS	CEA	Francis Belot francis.belot@cea.fr	KMS is a software package dedicated to the production management of a supercomputing center having a batch job queue manager and an accounting system. End users can manage/launch their simulation workflows and compute center production manager can control the workload on the systems managed by KMS.	Job management with workflow capabilities
Kosh	LLNL	Charles Doutriaux doutriaux1@llnl.gov	Leverages Sina as a data catalogue, and adds file movement/management, transparent file readers (into standard Python data structures like Numpy and Pandas), and utilities for machine learning workflows (e.g., sub-sampling, data subsetting for cross-validation).	Data Management
Maestro	LLNL	Frank Di Natale dinatale3@llnl.gov	A lightweight Python tool/library for constructing, running, and monitoring multi-step computational workflows specified in a straightforward YAML specification. Maestro is portable, with no infrastructure dependencies (other than the system scheduler) and makes it easy to repeat/share workflows.	Workflow manager
Merlin	LLNL	Luc Peterson peterson76@llnl.gov	Open source, distributed producer/consumer workflow system. Extends Maestro and provides a flexible task-graph to support many workflow motifs. Built on rabbitmq/redis/celery for high performance and proven scalability. Merlin supports workflows across multiple machines and batch jobs, providing a high throughput and low overhead capability. Merlin requires a Rabbit/MQ service for coordination.	Workflow manager
Next-Gen Workflow`	Sandia	Ernest Friedman-Hill ejfried@sandia.gov	NGW is a workflow system that includes a graphical workflow editor, an open document format, an extensible workflow engine, and a rich component library. It's designed to be used in a range of environments, from desktop to	Workflow

			HPC, and accommodates both fully automated and human-in-the-loop workflows.	
Parsl (part of ECP ExaWorks SDK)	ANL	Kyle Chard chard@uchicago.edu	Python library for parallel programming: Parsl provides an intuitive, pythonic way of parallelizing codes by annotating "apps": Python functions or external applications that run concurrently.	Workflow manager
PCVS	CEA	Julien Jaeger julien.jaeger@cea.fr	PCVS is a test engine designed to help users run large test bases in a scalable manner on HPC systems. It implements a powerfull test description model and embeds a high-throughput job sub-orchestrator. It also implements chaining primitives to build test workflows.	Test execution engine with workflow capabilities
PLATO	Sandia	Brett Clark bwclark@sandia.gov	PLATO is a topology optimization-based design environment. Using PLATO a designer can generate designs that are optimized to meet functional requirements. PLATO is made up of two parts: A user-friendly environment for problem setup, job submission, progress monitoring, and post processing, and a powerful optimization engine that can run locally or on massively parallel high performance computing platforms.	Workflow
RADICAL Cybertools	BNL	Shantenu Jha shantenu@bnl.gov	RADICAL Cybertools is an abstractions-based suite of workflow tools: SAGA abstraction layer for systems, Pilot for job allocations, EnsembleTK for ensemble workflows	Workflow Management Components
Sina	LLNL	Becky Haluska haluska1@llnl.gov	Sina consists of a C++ library for dumping non-bulk data in a common schema from applications, and a Python library for ingesting the data into SQL databases and querying with a simple API. Sina supports workflow orchestration by enabling simple queries and database updates. Sina supports data analysis by providing a data store that captures both meta-data and non-bulk data (scalars, vectors, time histories, ...).	Data Management

Themis	LLNL	David Domyancic domyancic1@llnl.gov	Themis is an ensemble manager that leverages existing user batch scripts, with minimal templating, combined with a CSV or similar representation of the variables to substitute, to scalably execute parameterized ensembles. Themis supports a CLI-only workflow and a Python API. In either case, dynamic workflows can be constructed in which new members can be defined asynchronously and added to the ensemble, enabling adaptive sample and optimization workflows.	Workflow manager